

***IN THE CLAIMS***

Claims 1-9 are cancelled herein. Claims 10-20 have been added. All pending claims are reproduced below.

1           1.       (Cancelled)

1           2.       (Cancelled)

1           3.       (Cancelled)

1           4.       (Cancelled)

1           5.       (Cancelled)

1           6.       (Cancelled)

1           7.       (Cancelled)

1           8.       (Cancelled)

1           9.       (Cancelled)

1           10.      (New) A method for array shape inferencing comprising the steps of:  
2                   determining an input shape-tuple for each operand of a program expression of an  
3                   array-based language;  
4                   analyzing the use of each operand in the program expression; and  
5                   determining a resulting shape-tuple of the program expression using an algebraic

6 framework.

1 11. (New) The method of claim 10, wherein the array based language is MATLAB.

1 12. (New) The method of claim 10, wherein determining a resulting shape-tuple of  
2 the program expression using an algebraic framework comprises the steps of:

3 determining a rank of the resulting shape-tuple; and,

4 promoting the input shape-tuples to an appropriate rank.

1 13. (New) The method of claim 12, wherein determining the rank of the resulting

2 shape-tuple comprises the steps of:

3 determining the ranks of each input operand;

4 identifying an operator corresponding to the input operands; and,

5 ascertaining the rank of the resulting shape-tuple according to the operator and the ranks

6 of the input operands.

1 14. (New) The method of claim 12, wherein promoting the input shape-tuples to an

2 appropriate rank comprises the steps of:

3 identifying the rank of the resulting shape-tuple;

4 expanding the input shape-tuples to correspond with the rank of the resulting shape-tuple;

5 and,

6 appending trailing extents of expanded input shape-tuples with an appropriate value.

1 15. (New) The method of claim 10, further comprising the steps of:

2 identifying a built-in function in the program expression;

3 determining a shape-tuple operator for the built-in function; and,

4 applying the operand shape-tuples to the shape-tuple operator for the built-in function.

1 16. (New) The method of claim 15, wherein determining a shape-tuple operator for  
2 the built-in function comprises the steps of:

3 identifying a shape-tuple expression corresponding to the built-in function;  
4 and, assigning the shape-tuple expression as the shape-tuple operator.

1 17. (New) The method of claim 16, further comprising the step of assigning a shape  
2 predicate to the resulting shape-tuple.

1 18. (New) The method of claim 10, further comprising the steps of:  
2 performing an array conformability check at run-time for a first statement;  
3 and  
4 applying a result of the conformability check to a second statement.

1 19. (New) The method of claim 18, further comprising the step of:  
2 determining a relationship among the first statement and the second statement.

1 20. (New) In the method of claim 18, further comprising the step of:  
2 preallocating a shape to a variable of a statement in a loop execution.